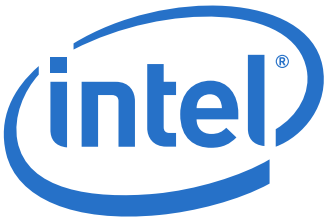

# Intel®Quark SoC X1000

## Specification Update

***October 2013***

**Notice:** The Intel®Quark SoC X1000 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Document Number: 329677-001US

# Revision History

| Date | Revision | Description |
|---|---|---|
| October 2013 | 001 | Initial revision |

# Contents



§ §

# Introduction

## Purpose/Scope/Audience

This document is an update to the specifications listed in the Parent Documents/ Related Documents table that follows. This document is a compilation of Errata, Specification Changes, Specification Clarifications, and Document-Only Changes. It is intended for hardware and software system designers and manufacturers as well as developers of applications, operating systems, or tools.

Information types defined in Conventions and Terminology are consolidated into the Specification Update and are no longer published in other documents.

This document may also contain information that was not previously published.

**Table 1. Parent Documents/Related Documents**

| Title | Number |
|---|---|
| *Intel®Quark SoC X1000 Datasheet* | Note 1 |
| *Intel®Quark SoC X1000 Platform Design Guide* | Note 1 |

***Notes:***
1. Contact your Intel sales representative. Some documents may not be available at this time.

## Conventions and Terminology

*Note:* Errata remain in the Specification Update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the Specification Update are archived and available upon request. Specification Changes, Specification Clarifications and Document-Only Changes are removed from the Specification Update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

**Table 2. Conventions and Terminology (Sheet 1 of 2)**

| Term | Definition |
|---|---|
| Document-Only Changes | Document-Only Changes are changes to an Intel Parent Specification that result in changes only to an Intel customer document but no changes to a specification or to a parameter for an Intel product. An example of a document-only change is the correction of a typographical error. |
| Errata (plural) Erratum (singular) | Errata are design defects or errors. These may cause the Clanton's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices. |

## Table 2. Conventions and Terminology (Sheet 2 of 2)

| Term | Definition |
|---|---|
| Parent Specification | A parent specification is a top-level specification from which other documents can be derived, depending on the product or platform. Typically, a parent specification includes a product's pinout, architectural overview, device operation, hardware interface, or electrical specifications.<br>Examples of parent specifications include the following: Datasheet, External Design Specification (EDS), Developer's Manual, Technical Product Specification.<br>The derived documents may be used for purposes other than that for which the parent specification is used. |
| Specification Changes | Specification Changes are the result of adding, removing, or changing a feature, after which an Intel product subsequently operates differently than specified in an Intel Parent Specification, but typically the customer does not have to do anything to achieve proper device functionality as a result of Intel adding, removing, or changing a feature. |
| Specification Clarifications | Specification Clarifications are changes to a document that arise when an Intel Parent Specification must be reworded so that the specification is either more clear or not in conflict with another specification. |

# Summary Tables of Current Product Issue Activity

Table 4 through Table 7 indicate the Errata, Specification Changes, Specification Clarifications, and Document-Only Changes that apply to the SoC product. Intel may fix some of the Errata in a future stepping of the component as noted in Table 3 or account for the other outstanding issues through Specification Changes, Specification Clarifications, or Document-Only Changes. Table 4 through Table 7 use the codes listed in Table 3.

**Table 3. Codes Used in Summary Tables**

| Code | Column | Definition |
|---|---|---|
| X | Stepping | Indicates either that, for the stepping/revision listed,<br>• an erratum exists and is not yet fixed<br>• a specification change or specification clarification applies |
| No mark or blank | Stepping | Indicates either that, for the stepping/revision listed,<br>• an erratum is fixed<br>• a specification change or specification clarification does not apply |
| Plan Fix | Status | This erratum may be fixed in a future stepping/revision. |
| Fixed | Status | This erratum has been previously fixed. |
| No Fix | Status | There are no plans to fix this erratum. |
| A change bar to the left of a table row indicates an item that is either new or modified from the previous version of the Specification Update document. | | |

## Table 4. Errata

| No. | Stepping/ Revision | Status | Errata Title |
|---|---|---|---|
| | A0 | | |
| 1 | X | No Fix | 4930600: High Speed SD Cards and eMMC Cards Fail to Initialize |
| 2 | X | No Fix | 4930644: PCIe Non-posted Transactions Can Overtake Posted Ones |
| 3 | X | No Fix | ESP Register Address Increment During POPA/POPAD Instruction Execution |
| 4 | X | No Fix | Missing Code Break Point |
| 5 | X | No Fix | Ucode Incorrectly Takes GPF on RDTSC Instruction in VM86 Mode with CPL=3 & CR4.TSD Not Set |
| 6 | X | No Fix | WBINVD Instruction Failed to Return Correct EIP |
| 7 | X | No Fix | Incorrect Processor State in V86/VME Mode |
| 8 | X | No Fix | Quark Core Incorrectly Updates ESP |
| 9 | X | No Fix | Fault Address is Not Loaded into CR2 |
| 10 | X | No Fix | #UB Exception not Detected in Protected Mode |
| 11 | X | No Fix | ESP Update Lost Following #UD |
| 12 | X | No Fix | EFLAGS.OF Incorrectly Updated in 16-bit Mode |
| 13 | X | No Fix | Possible Core Hang During Probe Mode if the SMM Region is Cached. |
| 14 | X | No Fix | Page Fault Not Detected on Exit from Probe Mode |
| 15 | X | No Fix | #GP Fault Incorrectly Generated in Real Mode |
| 16 | X | No Fix | Quark Core Fails to Generate #GP During a 4G Address Increment Wrap Around |
| 17 | X | No Fix | A Write to CR3 Does not Flush Prefetched Instructions |
| 18 | X | No Fix | #GP Fault When EIP Roll Over Occurs |
| 19 | X | No Fix | #PF Incorrectly Detected in Protected Mode |
| 20 | X | No Fix | Precision Bits in FCW are Set Incorrectly |
| 21 | X | No Fix | Erroneous Debug Breakpoint in PAE Mode.Will not be fixed |

## Table 5. Specification Changes

| No. | Stepping/ Revision | Specification Changes |
|---|---|---|
| | A0 | |
| | | None for this revision of the Specification Update. |

## Table 6. Specification Clarifications

| No. | Stepping/ Revision | Specification Clarifications |
|---|---|---|
| | A0 | |
| | | None for this revision of the Specification Update. |

## Table 7. Document-Only Changes

| No. | Document Title | Rev. | Document-Only Changes |
|---|---|---|---|
| | | | None for this revision of the Specification Update. |

# General Product Information

The Intel®Quark SoC X1000 SoC can be identified by the following register contents:

**Table 8. Identification Information**

| Part Number | Stepping | Vendor ID[1] | Device ID[2] | Revision Number[3] |
|---|---|---|---|---|
| Contact your Intel sales representative. | A0 | 8086 | 0958 | 00 |

***Notes:***
1. The Vendor ID corresponds to bits [15:0] of the VID - Vendor Identification Register located at Offset 00–01h in the PCI Function 0 configuration space.
2. The Device ID corresponds to bits [15:0] of the DID - Device Identification Register located at Offset 02–03h in the PCI Function 0 configuration space.
3. The Revision Number corresponds to bits [7:0] of the RID - Revision Identification Register located at Offset 08h in the PCI Function 0 configuration space.

The Intel®Quark SoC X1000 stepping can be identified by the following component markings:

| Stepping | Identifier | MM # | SPEC Code |
|---|---|---|---|
| A0 | Intel® Galileo Board X1000 | 929172 | ES |
| A0 | Intel® Galileo Board X1000 | 929184 | ES |
| A0 | Intel® Galileo Board X1000 | 929518 | ES |
| A0 | Intel® Galileo Board X1000 | 930236 | SS |
| A0 | Intel® Galileo Board X1000 | 930237 | SS |
| A0 | Intel® Galileo Board X1000 | 930239 | SS |

# Errata

## 1. 4930600: High Speed SD Cards and eMMC Cards Fail to Initialize

Problem: High Speed SD cards and eMMC cards fail to initialize correctly when connected to Quark X1000.

Implication: High Speed SD card or eMMC cards are not available for use in the system.

Workaround: A driver workaround exists to resolve this issue.

The driver tells the SD controller that the hi-speed SD card does not have a high speed bit. This workaround changes the high speed enable bit in the host control 1 register (28H) to always be set to 0.

This workaround is enabled by default in the sdhci driver within the Quark X1000 software release version 0.6.0 and later.

Status: No Fix

Please use the workaround detailed above.

## 2. 4930644: PCIe Non-posted Transactions Can Overtake Posted Ones

Problem: IPF bit in PCIe configuration register (D4h-MCP2), which is currently defaulting to '0', needs to be set to '1' in PCIe configuration.

Implication: PCIe non-posted transactions are passing out posted ones.

Workaround: See Appendix A.1 of UEFI FWG V0.5.0 or later for details.

The workaround is enabled in software release version 0.5.0 and later.

Status: No Fix

Firmware workaround should be used as detailed above.

## 3. ESP Register Address Increment During POPA/POPAD Instruction Execution

Problem: During POPA/POPAD instruction execution the contents of the stack are popped off the stack, if ESP is incremented as part of the operation, new address is not checked for page faults or breakpoints

Implication: Some page faults/Breakpoints for a specific code flow will be missed.

Workaround: None

Status: No Fix

## 4. Missing Code Break Point

Problem: When a conditional jump followed by jcc/short near jmp/near call and bit [2:0] of target address of second branch instruction is '0, code break point could potentially be missed.

Implication: Code break point set at addresses matching this code pattern might not be triggered.

Workaround: Set break point earlier in flow and step through each instruction.

Status: No Fix

### 5. Ucode Incorrectly Takes GPF on RDTSC Instruction in VM86 Mode with CPL=3 & CR4.TSD Not Set

Problem: The Quark core incorrectly takes GPF on RDTSC instruction in VM86 mode with CPL=3 & CR4.TSD not set.

Implication: User may experience spurious invalid General Purpose Fault.

Workaround: None

Status: No Fix

### 6. WBINVD Instruction Failed to Return Correct EIP

Problem: In 16-bit mode, the execution of the WBINVD instruction may not return the correct EIP value.

Implication: The WBINVD instruction may corrupt the contents of EIP in 16-bit mode.

Workaround: None

Status: No Fix

### 7. Incorrect Processor State in V86/VME Mode

Problem: The Quark Core incorrectly sets processor state in V86/VME mode on the RETD instr triggering a #GP trap.

Implication: Spurious General Purpose Fault.

Workaround: None

Status: No Fix

### 8. Quark Core Incorrectly Updates ESP

Problem: Processor incorrectly updates upper 16 bits of ESP during RETD instruction execution with privilege level change following ADD operation with result above 0xFFFFh.

Implication: ESP is corrupted during privilege mode change.

Workaround: None

Status: No Fix

### 9. Fault Address is Not Loaded into CR2

Problem: Following a double fault scenario, the second fault address is not loaded into CR2 register.

Implication: The source of the second fault address in a double fault scenario will not be accessible.

Workaround: None

Status: No Fix

### 10. #UB Exception not Detected in Protected Mode

Problem: In protected mode, #UD exception is not detected while accessing CR7 register during instruction MOV CR7, EBP.

Implication: #UD exception is not generated as expected in protected mode when doing a MOV CR7, r32.

Workaround: None

Status: No Fix

### 11. ESP Update Lost Following #UD

Problem: After an instruction updates the ESP value and the next inst. causes a #UD, the ESP update is lost.

Implication: Spurious corruption of the ESP register leading to corrupt code execution.

Workaround: None

Status: No Fix

### 12. EFLAGS.OF Incorrectly Updated in 16-bit Mode

Problem: In 16-bit mode, EFLAGS.OF is incorrectly updated on RCL instruction with zero count.

Implication: Corrupt code execution may occur in 16-bit mode.

Workaround: None

Status: No Fix

### 13. Possible Core Hang During Probe Mode if the SMM Region is Cached.

Problem: Core microcode flush after a Probe Mode save cycle will cause the core to hang if the SMM region is cached.

Implication: Core will hang on Probe mode exit.

Workaround: Do not cache System Management Mode memory. This is implemented by Intel's provided UEFI firmware.

Status: No Fix

### 14. Page Fault Not Detected on Exit from Probe Mode

Problem: The Probe Mode entry is currently the highest priority event and when a page fault happens in the same clock as the probe mode entry, the CR2 register is corrupted.

Implication: Page fault is not detected and on exit from probe mode, the CPU shall start executing from the wrong address.

Workaround: None

Status: No Fix

### 15. #GP Fault Incorrectly Generated in Real Mode

Problem: #GP Fault is incorrectly generated in Real Mode when executing the mov word ptr cs:[di + 0x48], cs instruction due to the stale Access Right bits when transitioning from Protected Mode to Real Mode.

Implication: Spurious general purpose fault may occur on transition from protected mode to real mode.

Workaround: None

Status: No Fix

### 16. Quark Core Fails to Generate #GP During a 4G Address Increment Wrap Around

Problem: On a 4G address increment wrap around, processor will fail to generate general purpose fault under certain circumstances.

Implication: Incorrect code execution and possible core hang for specific code flow and timing circumstances.

Workaround: None

Status: No Fix

### 17. A Write to CR3 Does not Flush Prefetched Instructions

Problem: Write to CR3 doesn't flush prefetched instruction when PAEXD is not enabled.

Implication: if CR3 value changes, prefetched instruction may use stale CR3 value, leading to corrupt code execution.

Workaround: None

Status: No Fix

### 18. #GP Fault When EIP Roll Over Occurs

Problem: #GP fault generated in case of EIP rollover in JO instruction with OF=0.

Implication: Spurious incorrect General purpose fault triggered.

Workaround: None

Status: No Fix

### 19. #PF Incorrectly Detected in Protected Mode

Problem: Incorrect detection of #PF by JNP instruction in Protected Mode.

Implication: Spurious protection fault generated.

Workaround: None

Status: No Fix

### 20. Precision Bits in FCW are Set Incorrectly

Problem: Whenever there is a GP fault on FLDENV, the precision bits are changed from 11 (double extended precision) to 00 (single precision).

Implication: Incorrect precision is set in the context of general purpose fault leading to incorrect code execution.

Workaround: None

Status: No Fix

### 21. Erroneous Debug Breakpoint in PAE Mode.

Problem: Debug Break Point triggered incorrectly when PDPTE entries are reloaded during task switching.

Implication: The system will spuriously enter probe mode. Recovery involves connecting a debugger to trigger a probe mode exit or a reboot.

Workaround: None

Status: No Fix



# Specification Changes

None for this revision of this Specification Update.

# Specification Clarifications

None for this revision of this Specification Update.



# Document-Only Changes

None for this revision of this Specification Update.

§ §